LAMP-TR-120
CS-TR-4698
UMIACS-TR-2005-07

MARCH 2005

HEADLINE GENERATION FOR WRITTEN AND BROADCAST NEWS

David Zajic, Bonnie Dorr. Richard Schwartz

Language and Media Processing Laboratory
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275
dmzajic, bonnie@umiacs.umd.edu
schwartz@bbn.com

Abstract

This technical report is an overview of work done on Headline Generation for written and broadcast news. The report covers HMM Hedge, a statistical approach based on the noisy channel model, Hedge Trimmer, a parse-and-trim approach using linguistically motivated trimming rules, and Topiary, a combination of Trimmer and Unsupervised Topic Discovery. Automatic evaluation of summaries using ROUGE and BLEU is described and used to evaluate the Headline Generation systems.

Keywords: Automatic Summarization, Automatic Evaluation of Summarization

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAR 2005**	2. REPORT TYPE	3. DATES COVERED **00-03-2005 to 00-03-2005**

4. TITLE AND SUBTITLE
Headline Generation for Written and Broadcast News

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Language and Media Processing Laboratory,Institute for Advanced Computer Studies,University of Maryland,College Park,MD,20742-3275

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**22**	

1. Introduction

This report presents work done from 2002 to 2005 for the TIDES project in the area of automatic headline generation. A headline is a very short (approximately 75 character) summary of a news story. Our goal is to produce *informative* abstracts that tell what happened in a news story. Other types of headlines include *indicative* headlines that tell what topic the story is about, and *eye-catcher* headlines that don't tell about the content of the story but are designed to entice people to read the story.

Our general approach is to select words in order from the story to form a headline. We allow for the possibility of morphological variation of individual words to preserve grammaticality. Section 2 discusses work we did to demonstrate the feasiblity of this approach for written news. Section 3 describes our first approach to the problem, HMM Hedge, a statistical approach that was biased to favor certain pragmatic linguistic characteristics.

Section 4 presents our second approach to the problem, Hedge Trimmer, which consists of linguistically-motivated sentence compression of lead sentences. In Section 5 we discuss topic lists, implemented by Unsupervised Topic Discovery, as an alternative to fluent headlines, and in Section 6 we describe Topiary, an approach that combines topic lists and fluent text to form more informative short summaries. Section 7 describes how Topiary was extended to cover the domain of transcripts of broadcast news.

The evaluation tools that we have used for intrinsic evaluation, BLEU and ROUGE, are described in Section 8, and the results of evaluations on written and broadcast news domains are given in Section 9. An extrinsic task for testing whether automatic headline generation (and short summaries in general) helps humans perform a task, or helps humans perform a task more quickly without loss of accuracy is discussed in Section 10.

2. Feasibility

Our approach is based on the selection of words from the original story, in the order that they appear in the story, and allowing for morphological variation. To determine the feasibility of our head-line-generation approach, we first attempted to apply our “select-words-in-order” technique by hand. We examined 73 stories from the TIPSTER corpus and found that it was possible to produce a fluent and accurate informative headline for all of the stories. Two researchers each constructed a head-line for each of the 73 stories, using words or morphological variants of words from the stories in order.

Of the 146 headlines, 2 did not meet the “select-words-in-order” criteria because of word reordering. We found that at least one fluent and accurate headline meeting the criteria was created for each of the stories. Further, we discovered that, with no instructions about sentence boundaries, the researchers constructed headlines entirely of words from the first sentence 80.1% of the time, and at a finer grain, 86.7% of the headline words were chosen in order from the first sentence. We conclude that our approach demonstrates promise for stories that are written as paragraphs of prose.

As part of this initial feasibility evaluation, we observed that only 8.9% of our 146 human-generated headlines used words beyond the first sentence, and none of the 144 valid headlines used words from beyond the fourth sentence.

The average length of the headlines was 10.76 words. Stories whose headlines required the later sentences tended to be human-interest stories with attention-grabbing introductions or they appeared to be excerpts from the middle of larger stories. Thus, in our current model, we adopt the additional constraint that headline words must be chosen from the first sentence of the story, using a threshold of 10 headline words.

3. HMM Hedge Algorithm Description

Our algorithm for selecting story words to form headlines is based on a standard “noisy channel” model of processing-with a subsequent decoder for producing headline words from stories.

The intuition is to treat stories and headlines as the joint output of a generative model. Our approach is to find the headline most likely to have been generated jointly with a given story. In a given story, some words will be identified as headline words. The headline will be composed of the headline words, or morphological variants of the headline words. Thus, stories consist of headline words with many other words interspersed amongst them, and the most likely headline is determined by calculating the most likely set of headline words given that the observed story was generated. Thus, stories consist of headline words (or morphological variants of headline words) with many other words interspersed amongst them.

Formally, if H is a ordered subset of the first N words of story S, we want to find the H which maximizes the likelihood that H is the set of headline words in story S, or:

$$\mathrm{argmax}_H\ P(H|S)$$

It is difficult to estimate P(H|S), but this probability can be expressed in terms of other probabilities that are easier to compute, using Bayes' rule:

$$P(H|S)= [P(H)P(S|H)] / \{P(S)$$

Since the goal is to maximize this expression over H, and P(S) is a constant with respect to H, P(S) can be omitted. Thus we wish to find:

$$\mathrm{argmax}_H\ P(H)P(S|H)$$

3.1 Source Model: Bigram Estimates of Headline Probabilities

We estimate P(H) using the bigram probabilities of the headline words used in the story:

$$P(H) = P(h_1|start)P(h_2|h_1)...p(h_n|end)$$

3.2 Generative Model: Using HMMs for Story Generation from Headlines

To estimate P(S|H) we must consider the process by which a story is generated. This process can be represented as a Hidden Markov Model (HMM). A HMM is a weighted finite-state automaton in which each state probabilistically emits a string. The simplest HMM to generate stories with headlines is shown in Figure 1.

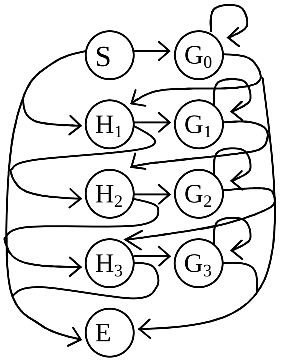

Figure 1: HMM for Three-Word Story

Consider the following story.

(1) Story Words: After months of debate following the Sept. 11 terrorist hijackings, the Transportation Department has decided that airline **pilots** will **not** be **allowed to have guns in** the **cockpits**.

Generated Headline: Pilots not allowed to have guns in cockpits

The H state will emit the words in bold (pilots, not, allowed, to, have, guns, in, cockpits), and the G state will emit all the other words. The HMM will transition between the H and G states as needed to generate the words of the story.

We use a unigram model of stories and a bigram model of headlines based on a corpus of 496215 stories from Associated Press, Wall Street Journal and San Jose Mercury News. Because of the bigram model of the headline language, the HMM in Figure 1 will not be sufficient. The HMM for a three-word story is shown in Figure 1 above. There should be an H state in the HMM for each word in the headline vocabulary. Since we can observe the words in the story, it is sufficient to have an H state for each word in the story. Each H state will have a corresponding G state which emits story words until the next headline word and remembers the previous emitted headline word.

The HMM starts in start state S. It can transition from S to any H state or to state G0. When the HMM is in an H state it emits a headline word. From an H state, the HMM can transition to any later H state or to the corresponding G state. From any G state, the HMM can stay in that state or transition to any later H state. Any state can transition to the end state E.

Suppose we observe the story in (1) as the output of an HMM. There are 28 words in that story, so there will be 28 H states, 29 G states, a start state S and an end state E in the HMM. The headline in (1) can generate the story as follows. The HMM will start in state S, emit a start symbol and transition to state G_0. It will stay in G_0 and emit the words *after*, *months*, *of*, *debate*, ..., *decided*, *that* and *airline*. Then it will transition to state H_{pilots} and emit the word *pilots*.

The next word in the story is not a headline word, so the HMM transitions to the corresponding G state, G_{pilots}, which emits *will*. Note that being in state G_{pilots} allows the machine to remember that pilots is the last emitted headline word. The next story word is a headline word, so we transition to H_{not} and emit *not*. Skipping ahead to after $H_{allowed}$ has emitted *allowed*, we note that the next story word is also a headline word. In this case, the HMM does not go into the corresponding G state, but instead goes directly to H_{to}.

Transitions from H states to a later H state corresponds to a clump of sequential headline words in the story. A transition from a H state to a G state corresponds to the end of a clump and the start of a gap, i.e., a headline word followed by non-headline word.

Conversely, a transition from a G state to a H state corresponds to the end of a gap and the start of a clump. Finally, after cockpits is emitted by $H_{cockpits}$, the HMM goes to the end state. If there had been more words in the story after cockpits, they would all be emitted by $G_{cockpits}$, then the HMM would go to the end state.

This process can be thought of as one in which a story and a headline are generated simultaneously. Alternatively, we can think of the headline as the input to an HMM controlling the sequence of H states, but in which the model is free to transition to G states at any time. This view fits the Noisy Channel Model interpretation.

P(S|H) is estimated using this HMM. The H states can only emit their specific word from the headline vocabulary with probability 1. The G states can emit any word w in the general language vocabulary with probability P(w).

Every possible headline corresponds to a path through the HMM which successfully emits the story. The path through the HMM described above is not the only one that could generate the story in (1). Other possibilities are:

(2) Transportation Department decided airline pilots not to have guns

(3) Months of the terrorist has to have cockpits

Although (2) and (3) are possible headlines for (1), the conditional probability of (3) given (1) will be lower than the conditional probability of (2) given (1).

3.3 Viterbi Decoding

We use the Viterbi algorithm to select the most likely headline for a story. The implementation took advantage of the constraints that we imposed on headlines: that headline words are taken from the story in the order that they appear. Headline states can only emit a specific word, and all other words have zero probability. Each headline state has transitions only to the following headline state or to the corresponding G state.

3.4 Decoding Parameters

In the course of our investigation, we added four decoding parameters motivated by intuitive observations of the output. Our goal was to make the results more like Headlinese. The decoding parameters are

- a length penalty
- a position penalty
- a string penalty
- a gap penalty

Note that the incorporation of these parameters changes the values in the cells from log probabilities to relative desirability scores.

We tested different values of the four parameters by trial and error. A logical extension to this work would be to attempt to learn the best setting of these parameters, e.g., through Expectation Maximization.

3.4.1 Length Penalty

The most salient parameter is the length penalty. We have observed that headlines are usually 5 to 15 words long. The initial translation model had no pressure for headlines in this length range. It is possible for the algorithm to generate headlines of length N which include all the story words, or of length zero.

The length penalty biases the algorithm towards shorter or longer headlines as follows. The transition probability from a G state to itself is multiplied by the length penalty. A length penalty greater than one will favor paths which spend more time in G states, and thus have fewer headline words. A length penalty less than one will favor paths which spend less time in G states, and thus have more headline words. The goal is to nudge the headline length into a specific length range, so no single length penalty is suitable for every story. We iterate the Viterbi algorithm, adjusting the length penalty until the headline length falls in the desired range.

3.4.2 Position Penalty

We observed that, in the human-constructed headlines, the headline words tended to appear near the front of the story. The position penalty is used to favor headline words from early in the story. The story word in the nth position is assigned a position penalty of pn, where p is a positive number less than one. The emission probabilities on H states are multiplied by the position penalty for the position of the word being considered. Thus words near the front of the story carry less of a position penalty than words farther along. This technique doesn't work in the case of human interest and sports stories that start with a hook to get the reader's attention, before getting to the main topic of the story.

3.4.3 String Penalty

We observed that the human-constructed headlines often contained contiguous strings of story words in the headlines. Examples (1) and (2) above illustrate this with strings such as “allowed to have guns,” and “embargo against Cuba.” The string penalty is used as a bias for “clumpiness”, i.e., the tendency to generate headlines composed of strings of contiguous story words. Each transition from an H state to its G state is multiplied by the string penalty. A string penalty lower than one will cause the algorithm to prefer clumpy headlines.

3.4.4 Gap Penalty

Very large gaps between headline words tend to be a sign of great effort from the human to piece together a headline from unrelated words. We believe that the algorithm would not be nearly as successful as the humans in constructing large gap headlines, and that allowing it to try would cause it to miss easy, non-gappy headlines.

The gap penalty is used to bias against headline “gappiness”, i.e., the tendency to generate headlines in which contiguous headline words correspond to widely separated story words. At each transition from a G state to a H state, a gap penalty is applied which depends on the size of the gap since the last headline word was emitted. This can also be seen as a penalty for spending too much time in one G state. Low gap penalties will cause the algorithm to favor headlines with few large gaps.

3.4.5 Morphological Extensions to the Model

Headlines usually use verbs in the present tense while stories use verbs in the past tense. This observation suggests that our initial algorithm omitted important content words from headlines because their probability in the headline language model is low.

The algorithm has been modified to accommodate morphological variations as follows. Each story word is expanded into its set of morphological variants, such that each story-position is associated with a set of strings. In the HMM there is a H state and a G state for each story-position string pair. For example, if the second word in the story is "said", there can be an H state capable of emitting "says," and the word "says" can appear in the generated headline.

The emission probability for an H state is nonzero for all the morphological variants of that word. At present this probability is 1/n, where n is the number of morphological variants. In future work, this will be biased in favor of the morphological variations that are observed between headlines and stories.

4. Trimmer Algorithm Description

In this section we will give an overview of how we create short summaries for written news stories and broadcast news stories, placing Hedge Trimmer and UTD in that context. We will describe Hedge Trimmer and UTD and how they are combined to produce Topiary. We will discuss some issues involved in extending Topiary to the domain of broadcast news.

4.1 Overview

We use a parse-and-trim approach to produce short summaries for written news stories. We select a lead sentence from the story, parse the lead sentence, and add named-entity information to the parse. We trim the parse tree by pruning constituents according to a linguistically-motivated heuristic until the surface form of the parse is below the desired length threshold. We then produce a list of topics and augment the trimmed result with a subset of these topics using a dynamic threshold on the overall length. Each of these processes is described in detail below.

4.2 Hedge Trimmer

We have designed and implemented an algorithm called Hedge Trimmer that generates a headline for a news story by compressing the *lead* (or main) topic sentence according to a linguistically-motivated algorithm.[1] The compression consists of parsing the sentence using the BBN SIFT parser and removing low-content syntactic constituents.

Low-content constituents are removed one-by-one until a length threshold has been reached. These include, among others, certain determiners (the, a), time expressions, relative clauses, verb-phrase conjunction, preposed adjuncts, *have* and *be* in past or present participle constructions, certain modal verbs, *it* in extraposition (*It was the third maritime accident*), *there* in existential clauses (*There has been a maritime accident*), and prepositional phrases that do not contain named entities.[2] These constituents are considered to be low-content because they occur less frequently in human summaries than in general text.

[1] Pre-processing for written news stories includes removal of datelines and other non-story text, such as instructions to editors.

[2] The named-entity tagging is done with BBN IdentiFinder (Bikel et al., 1999).

If Hedge Trimmer is unable to reach the desired threshold by removing these low-content constituents, we consider the removal of high-content constituents, such as adverbs, adverbial phrases, adjectives, adjective phrases and nouns that modify other nouns. If Hedge Trimmer is unable to reach the threshold even by removing high-content constituents, the trimmed sentence will be truncated to meet the length constraint. The threshold can be specified either in number of words or number of characters.

4.3 Trimming Rules

The input to Hedge is a story, whose first sentence is immediately passed through the BBN parser. The parse-tree result is passed to a linguistically-motivated module that selects story words to form headlines by means of three simple steps:

1. Choose lowest leftmost S with NP,VP
2. Remove low content units
    - some determiners
    - time expressions
3. Iterative shortening:
    - XP Reduction
    - Remove preposed adjuncts
    - Remove trailing PPs
    - Remove trailing SBARs

We discuss each of these three steps in turn.

STEP 1: Choose the Correct S Node

The first step relies on what is referred to as the *Projection Principle* in linguistic theory (Chomsky, 1981): Predicates project a subject (both dominated by S) in the surface structure. Our human-generated headlines generally conformed to this rule; thus, we adopted it as a constraint in our algorithm.

An example of the application of step 1 above is the following, where boldfaced material from the parse tree representation is retained and italicized material is eliminated:

(4) Input: Rebels agree to talks with government officials said Tuesday.

(5) Parse: *[S* **[S [NP Rebels] [VP agree to talks with government]]** *officials said Tuesday.]*

(6) Output of step 1: Rebels agree to talks with government.

When the parser produces a correct tree, this step provides a grammatical headline. However, the parser often produces an incorrect output. Human inspection of our 624-sentence DUC-2003 evaluation set revealed that there were two such scenarios, illustrated by the following cases:

[S [SBAR What started as a local controversy] [VP has evolved into an international scandal.]]

[NP [NP Bangladesh] [CC and] [NP [NP India] [VP signed a water sharing accord.]]]

In the first case, an S exists, but it does not conform to the requirements of step 1. This occurred in 2.6% of the sentences in the DUC-2003 evaluation data. We resolve this by selecting the lowest leftmost S, i.e., the entire string “What started as a local controversy has evolved into an international scandal” in the example above.

In the second case, there is no S available. This occurred in 3.4% of the sentences in the evaluation data. We resolve this by selecting the root of the parse tree; this would be the entire string “Bangladesh and India signed a water sharing accord” above. No other parser errors were encountered in the DUC-2003 evaluation data.

STEP 2: Removal of Low Content Nodes

This step of our algorithm eliminates low-content units. We start with the simplest low-content units: the determiners *a* and *the.* Beyond these, we found that the human-generated headlines did not include time expressions which, although certainly not content-free, do not contribute toward conveying the overall “who/what content” of the story. Since our goal is to provide an informative headline (i.e., the action and its participants), the identification and elimination of time expressions provided a significant boost in the performance of our automatic headline generator.

We identified time expressions in the stories using BBN’s Identifinder (Bikel et al, 1999). We implemented the elimination of time expressions as a two-step process:

- Use Identifinder to mark time expressions
- Remove [PP … [NP [X] …] …] and [NP [X]] where X is tagged as part of a time expression

The following examples illustrate the application of this step:

(7) Input: The State Department on Friday lifted the ban it had imposed on foreign fliers.

Parse: *[Det The]* **State Department** *[PP [IN on] [NP [NNP Friday]]]* **lifted** *[Det the]* **ban it had imposed on foreign fliers.**

Output of step 2: State Department lifted ban it has imposed on foreign fliers.

(8) Input: An international relief agency announced Wednesday that it is withdrawing from North Korea.

Parse: *[Det An]* **international relief agency announced** *[NP [NNP Wednesday]]* **that it is withdrawing from North Korea.**

Output of step 2: International relief agency announced that it is withdrawing from North Korea.

We found that 53.2% of the stories we examined contained at least one time expression which could be deleted. A human inspection of 50 deleted time expressions showed that 38 were desirable deletions, 10 were locally undesirable because they introduced an ungrammatical fragment,[3] and 2 were undesirable because they removed a potentially relevant constituent. However, even an undesirable deletion often pans out for two reasons: (1) the ungrammatical fragment is frequently deleted later by some other rule; and (2) every time a constituent is removed it makes room under the threshold for some other, possibly more relevant constituent. Consider the following examples.

[3] Two examples of genuinely undesirable time expression deletion are:

- The attack came on the heels of [New Year’s Day].
- [New Year’s Day] brought a foot of snow to the region.

(9) At least two people were killed Sunday.

(10) At least two people were killed when single-engine airplane crashed.

Example (9) was produced by a system which did not remove time expressions. Example (10) shows that if the time expression Sunday were removed, it would make room below the 10-word threshold for another important piece of information.

STEP 3: Iterative Shortening

The final step, iterative shortening, removes linguistically peripheral material—through successive deletions—until the sentence is shorter than a given threshold. We took the threshold to be 10 for the DUC task, but it is a configurable parameter. Also, given that the human-generated headlines tended to retain earlier material more often than later material, much of our iterative shortening is focused on deleting the rightmost phrasal categories until the length is below threshold.

There are four types of iterative shortening rules. The first type is a rule we call "XP-over-XP," which is implemented as follows:

> In constructions of the form [XP [XP …] …] remove the other children of the higher XP, where XP is NP, VP or S.

The motivation for this rule is that the human-produced headlines primarily include head words with short lexical modifiers, not phrasal-level modifiers. The rule is applied iteratively, from the deepest rightmost applicable node backwards, until the length threshold is reached.

The impact of XP-over-XP can be seen in these examples of NP-over-NP, VP-over-VP, and S-over-S, respectively:

(11) Input: A fire killed a firefighter who was fatally injured as he searched the house.

Parse: **[S** *[Det A]* **fire killed** *[Det a]* **[NP [NP firefighter]** *[SBAR who was fatally injured as he searched the house]* **]]**

Output of NP-over-NP: fire killed firefighter

(12) Input: Illegal fireworks injured hundreds of people and started six fires.

Parse: **[S Illegal fireworks [VP [VP injured hundreds of people]** *[CC and] [VP started six fires]* **]]**

Output of VP-over-VP: Illegal fireworks injured hundreds of people

(13) Input: A company offering blood cholesterol tests in grocery stores says medical technology has outpaced state laws, but the state says the company doesn't have the proper licenses.

Parse: **[S** *[Det A]* **company offering blood cholesterol tests in grocery stores says [S [S medical technology has outpaced state laws],** *[CC but] [S [Det the] state stays [Det the] company doesn't have [Det the] proper licenses.]]* **]**

Output of S-over-S: Company offering blood cholesterol tests in grocery store says medical technology has outpaced state laws

The second type of iterative shortening is the removal of preposed adjuncts. The motivation for this type of shortening is that the human-generated headlines tend to ignore what we refer to as the *preamble* of the story. Assuming the Projection principle has been satisfied, the preamble is viewed as the phrasal material occurring before the subject of the sentence. Thus, adjuncts are identified linguistically as any XP unit preceding the first NP (the subject) under the S chosen by step 1. This type of phrasal modifier is invisible to the XP-over-XP rule, which deletes material under a node only if it dominates another node of the same phrasal category.

The impact of this type of shortening can be seen in the following example:

(14) Input: According to a now finalized blueprint described by U.S. officials and other sources, the Bush administration plans to take complete, unilateral control of a post-Saddam Hussein Iraq

Parse: **[S** *[PP According to a now-finalized blueprint described by U.S. officials and other sources] [Det the]* **Bush administration plans to take complete, unilateral control of** *[Det a]* **post-Saddam Hussein Iraq]**

Output of Preposed Adjunct Removal: Bush administration plans to take complete unilateral control of post-Saddam Hussein Iraq

The third and fourth types of iterative shortening are the removal of trailing PPs and SBARs, respectively:

- Remove PPs from deepest rightmost node backward until length is below threshold.
- Remove SBARs from deepest rightmost node backward until length is below threshold.

These rules are applied with a backoff option to avoid over-trimming the parse tree. First the PP shortening rule is applied. If the threshold has been reached, no more shortening is done. However, if the threshold has not been reached, the system reverts to the parse tree as it was before any PPs were removed, and applies the SBAR shortening rule. If the threshold still has not been reached, the PP rule is applied to the result of the SBAR rule.

Other sequences of shortening rules are possible. The one above was observed to produce the best results on a 73-sentence development set of stories from the TIPSTER corpus. The intuition is that, when removing constituents from a parse tree, it's best to remove smaller portions during each iteration, to avoid producing trees with undesirably few words. PPs tend to represent small parts of the tree while SBARs represent large parts of the tree. Thus we try to reach the threshold by removing small constituents, but if we can't reach the threshold that way, we restore the small constituents, remove a large constituent and resume the deletion of small constituents.

The impact of these two types of shortening can be seen in the following examples:

(15) Input: More oil-covered sea birds were found over the weekend.

Parse: [S **More oil-covered sea birds were found** *[PP over the weekend]*]

Output of PP Removal: More oil-covered sea birds were found.

(16) Input: Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition while assuring closer cooperation after Hong Kong returns.

Parse: [S **Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition** *[SBAR while assuring closer cooperation after Hong Kong returns]*]

Output of SBAR Removal: Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition

Recently we have investigated a rendering of the summary as "Headlinese" (Mårdh, 1980) in which certain constituents are dropped with no loss of meaning. The result of this investigation has been used to enhance Hedge Trimmer, most notably the removal of certain instances of *have* and *be*. For example, the previous headline generator produced summaries such as Sentence (18), whereas the have/be removal produces Sentence (19).

(17) Input: At least 231 people have been confirmed dead in Honduras from former-hurricane Mitch, bringing the storm's death toll in the region to 357, the National Emergency Commission said Saturday

(18) Without participle have/be removal: At least 231 people have been confirmed dead bringing storm's death toll

(19) With participle have/be removal: At least 231 people confirmed dead in Honduras bringing storm's death toll

Have and *be* are removed if they are part of a past or present participle construction. In this example, the removal of *have been* allows a high-content constituent *in Honduras* to fit into the headline.

The removal of forms of *to be* allows Hedge Trimmer to produce headlines that concentrate more information in the allowed space. The removal of forms of *to be* results in sentences that are not grammatical in general English, but are typical of Headlinese English. For example, Sentences (21), (22) and all other examples in this paper were trimmed to fit in 75 characters.

(20) Input: Russian space experts were making final preparations Thursday at the Baikonur rocket base to launch the first component of a multibillion dollar international space station after a year of delay.

(21) Without participle to-be removal: Russian space experts were making final preparations

(22) With participle to-be removal: Russian space experts making final preparations at Baikonur rocket base

When *have* and *be* occur with a modal verb, the modal verb is also removed. Sentence (25) shows an example of this. It could be argued that by removing modals such as *should* and *would* the meaning is vitally changed. The intended use of the headline must be considered. If the headlines are to be used for determining query relevance, removal of modals may not hinder the user while making room for additional high-content words may help.

(23) Input: Famine-threatened North Korea's harvest will be no better this year than last and could be worse, a senior U.N. aid official said Saturday.

(25) Without modal-have/be removal: Famine threatened North Korea's harvest will be no better this year

(25) With modal-have/be removal: Famine threatened North Korea's harvest no better this year than last

In addition when *it* or *there* appears as a subject with a form of *be* or *have*, as in extraposition (*It was clear that the thief was hungry*) or existential clauses (*There have been a spate of dog maulings*), the subject and the verb are removed.

Finally, for situations in which the length threshold is a hard constraint, we added some emergency shortening methods which are only to be used when the alternative is truncating the headline after the threshold, possibly cutting the middle of a word. These include removal of adverbs and adverbial phrases, adjectives and adjective phrases, and nouns that modify other nouns.

The main benefit of have/be removal is that it often shortens a headline by five to eight characters, without losing any content and rarely causing the sentence to become ungrammatical as Headlinese. Sometimes this shortening is enough to allow another constituent or, as we discuss in Section 6, an additional topic word or phrase to fit under the length threshold.

5. Unsupervised Topic Discovery

Unsupervised Topic Discovery (UTD) (Schwartz et al., 2001) is used when we do not have a corpus annotated with topics. It takes as input a large unannotated corpus in any language and automatically creates a set of topic models with meaningful names. The algorithm has several stages. First, it analyzes the corpus to find strings of words that occur frequently. (It does this using a Minimum Description Length criterion.) These strings are frequently meaningful topic names.

Second, it finds the high-content words in each document (using a modified tf.idf measure). These are possible topic names for each document. It keeps only those names that occur in at least four different documents. [4] These are taken to be an initial set of topic names.

In the third stage UTD trains topic models corresponding to these topic names. The modified EM procedure of OnTopic is used to determine which words in the documents often signify these topic names. This produces topic models.

Fourth, these topic models are used to find the most likely topics for each document. This often adds new topics to documents, even though the topic name did not appear in the document.

We found, in various experiments, that the topics derived by this procedure were usually meaningful and that the topic assignment was about as good as when the topics were derived from a corpus that was annotated by people. We have also used this procedure on different languages and shown the same behavior.

Sentence (26) is a topic list generated for a story about the investigation into the bombing of the U.S. Embassy in Nairobi on August 7, 1998.

(26) BIN_LADEN EMBASSY BOMBING POLICE OFFICIALS PRISON HOUSE FIRE KABILA

6. Combination of Hedge Trimmer and Topics: Topiary

[4] The number of documents in which a name must occur is configurable. One might wish to use a number smaller than four for corpora in which there might not actually be four documents about a particular topic.

The Hedge Trimmer algorithm is constrained to take its headline from a single sentence. It is often the case that there is no single sentence that contains all the important information in a story. The information can be spread over two or three sentences, with pronouns or ellipsis used to link them. In addition, our algorithms do not always select the ideal sentence and trim it perfectly.

On the other hand, topics alone also have drawbacks. UTD rarely generates any topic names that are verbs. Thus topic lists are good at indicating the general subject are but rarely give any direct indication of what events took place.

Topiary is a modification of the Hedge Trimmer algorithm that takes a list of topics with relevance scores as additional input. The compression threshold is lowered so that there will be room for the highest scoring topic terms that aren't already in the headline. This amount of threshold lowering is dynamic, because the trimming of the sentence can remove a high-scoring topic term from the trimmed output, making that topic term eligible to be part of the topic list.

After trimming is complete, additional topic terms that do not occur in the headline are added to use up any remaining space. The number or requested topics is a parameter. If Hedge Trimmer cannot reach the adjusted threshold required for the requested number of topics, the Hedge Trimmer output will be truncated to ensure the required number of topics.

In experiments on written news, we found that the best ROUGE scores were achieved by requiring one topic. The Hedge Trimmer algorithm usually undershoots the threshold by a few characters, so this resulted in two topic terms per headline, on average.

Topiary produces a short summary consisting of one or more main topics about the story and a short sentence that says what happened concerning them. To aid in comprehension when the output is intended for human readers we present the topics in all caps and separate topic list from the fluent summary by a colon. The output submitted to automatic evaluation omits these conventions. The combination is often more concise than a fully fluent sentence and compensates for the fact that the topic and the description of what happened to it do not appear in the same sentence in the original story.

Sentences (27) and (28) are the output of Hedge Trimmer and Topiary for the same story for which the topics in Sentence (26) were generated.

(27) FBI agents this week began questioning relatives of the victims

(28) BIN_LADEN EMBASSY BOMBING:FBI agents this week began questioning relatives

Topiary was submitted to the Document Understanding Conference (DUC) Evaluation forum. Figure 2 shows how Topiary peformed in comparison with other DUC2004 participants on task 1, using ROUGE. Task 1 was to produce a summary for a single news document no more than than 75 characters long. The large open diamonds show the performance of Topiary according to six different variants of ROUGE. The filled diamonds are the other DUC2004 participants, and the open squares are the human references summaries. The dotted lines (and solid line for Topiary) connect scores from the same source. The different ROUGE variants are sorted along the horizontal axis by overall performance of the systems.

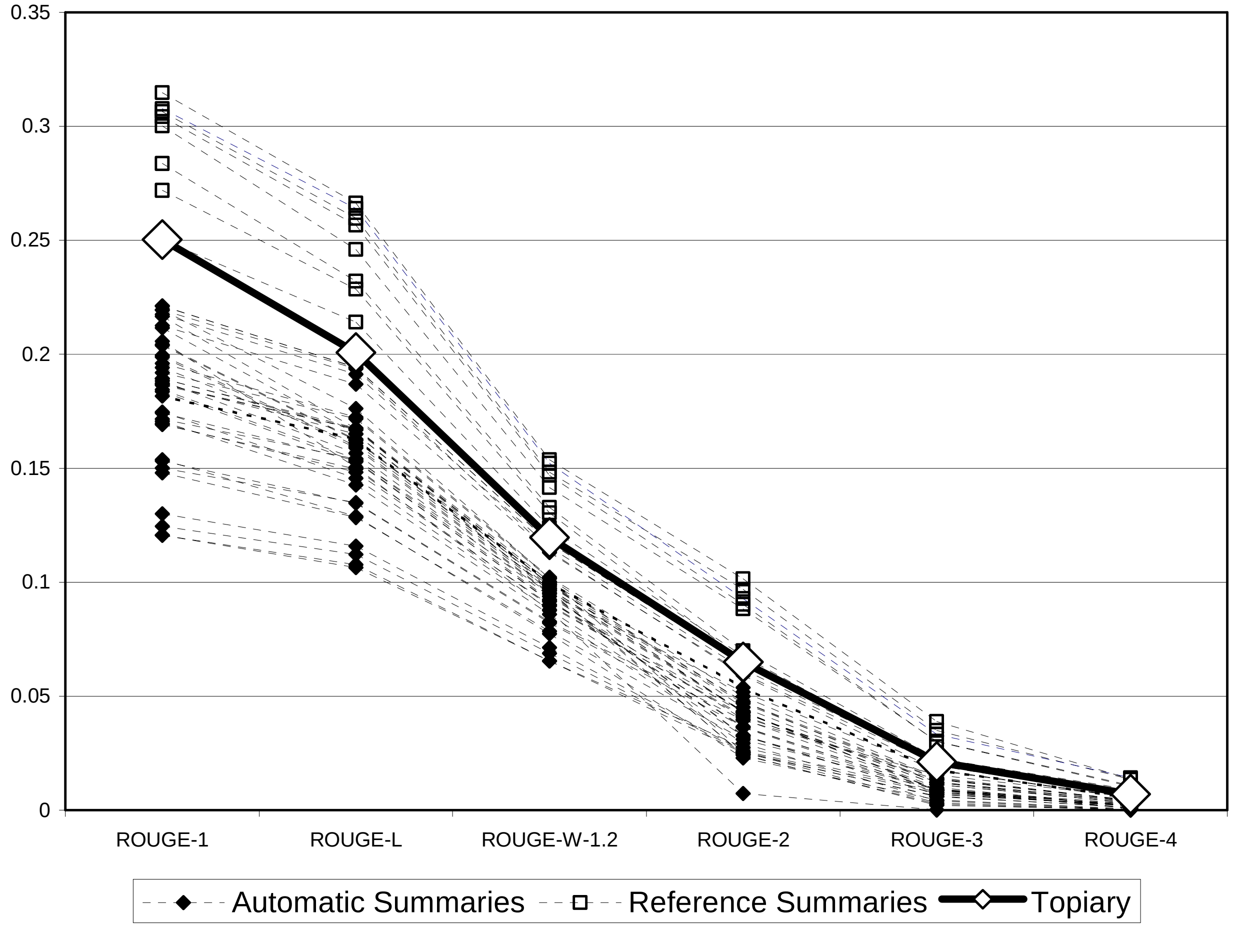

Figure 2: ROUGE Scores for DUC2004 Automatic Summaries, Reference Summaries and Topiary

The key observations are that there was a wide range of performance among the submitted systems, and that Topiary scored first or second among the automatic systems on each ROUGE measure.

7. Extension to Broadcast News

Our goal was to determine whether Topiary's design was as readily applicable to broadcast news as it was to text and, if not, what design changes would be necessary to broaden its applicability. Broadcast news stories require considerable pre-processing than written news. In order to reach a point where our summarization approach would be applicable to broadcast news, the news story had to go through Automatic Speech Recognition (ASR), story boundary detection and sentence boundary detection.

We used BBN BYBLOS News Transcription System (Ma et al., 2004) for ASR. Story boundary detection was done manually, and sentence boundary detection was done using the SRILM toolkit (Stolcke, 2002). We consider story boundary detection and sentence boundary detection to be outside of the scope of this work, although dealing appropriately with the noise introduced by these systems is within scope.

The pre-processing used for written news, parsing and named-entity tagging is also required for broadcast news. We use the same tools for these tasks, but with language models that were trained on broadcast news.

For written news stories, the first sentence of the document is taken to be the lead sentence. For broadcast news this is less often true. Broadcast news stories often begin with introductory or connective sentences, such as “This is CNN news,” or “Now we turn to international news.” Errors in story boundary detection can make the situation worse.

We used the M5' learning algorithm implemented by Weka (Witten and Frank 2000) to train a decision tree for lead sentence selection. However, this algorithm does not yet outperform the baseline of choosing the first sentence as the lead sentence. In the evaluations that follow, the first sentence of broadcast news stories will be taken to be the lead sentence, and improvement of the lead sentence chooser will be a high priority.

In applying the Hedge Trimmer algorithm to broadcast news we found that some of the linguistic triggers we used are more typical of written news than broadcast news. For instance, one such trigger was intended to catch attributions such as “The capital city is now under the control of rebel forces, *government officals announced.*” In broadcast news we observed that this information is more likely to be structured, “*Officals announced that* the capital city is now under the control of rebel forces,” and that the structure we had prepared for rarely occured. Further, with broadcast news we are more likely to see fragmented or incorrect parses.

Sentences (29) through (31) are actual outputs from Topiary on broadcast news stores from the TDT corpus, as transcribed by BYBLOS. These examples are atypically informative and coherent, however they show the promise of this approach.

(29) CASEY_MARTIN GOLF DISABLED:he's attracted attention because he's suing pga

(30) ALGERIA:the state department issued its annual report on human rights today

(31) IVORY_COAST:capital being torn apart massive demonstrations drove military

8. Evaluation Tools and Corpora

We used two automatic evaluation systems, BLEU (Papineni et al. 2002) and ROUGE (Lin and Hovy, 2003), to evaluate three variants of our headline generation systems on both written news and broadcast news. Both measures make n-gram comparisons of the candidate systems to a set of reference summaries.

8.1 ROUGE

ROUGE is a recall-based measure for summarizations. This automatic metric counts the number of n-grams in the reference summaries that occur in the candidate and divides by the number of n-grams in the reference summaries. The size of the n-grams used by ROUGE is configurable. ROUGE-*n* uses 1-grams through *n*-grams. ROUGE-L is based on longest common subsequences, and ROUGE-W-1.2 is based on weighted longest common subsequences with a weighting of 1.2 on consecutive matches of length greater than 1.

8.2 BLEU

BLEU is a system for automatic evaluation of machine translation that uses a modified n-gram precision measure to compare machine translations to reference human translations. This automatic metric counts the number of n-grams in the candidate that occur in any of the reference summaries and divides by the number of n-grams in the candidate. The size of the n-grams used by BLEU is configurable. BLEU-*n* uses 1-grams through *n*-grams. In our evaluation of headline generation systems, we treat summarization as a type of translation from a verbose language to a concise one, and compare automatically generated headlines to human generated headlines.

8.3 Corpora and References

For both written news and broadcast news we ran three summarization systems: Topiary, Hedge Trimmer and UTD. The corpus for the written news was 489 stories from the DUC2004 single-document summarization headline generation task. We used the four references per document that were provided by NIST for the DUC2004 evaluation. Topiary, Hedge Trimmer and UTD were run on this corpus, with a length threshold of 75 characters. For Topiary and Hedge Trimmer the first sentence of the document was taken to be the lead sentence. Topiary was configured to require one topic. We created a baseline consisting of the first 75 characters of each story.

The average output sizes of each system are shown in 1. Hedge Trimmer headlines tend to be shorter than the threshold because Hedge Trimmer removes constituents until the length is below the threshold. Sometimes it must remove a large constituent in order to get below the threshold. Topiary is able to make full use of the space by filling in topic words.

System	Description	Words	Chars
Topiary	Combination of UTD and H. Trimmer	10.7	73.2
H. Trimmer	H. Trimmer alone	8.6	57.4
UTD	UTD topics alone	9.5	71.1

Table 1: Systems and Average Summary Lengths for Written News

The corpus for the broadcast news consisted of 560 broadcast news stories from the TDT corpora. We commissioned two reference summaries for each story for use in this evaluation. Topiary, Hedge Trimmer and UTD were run on this corpus using the same configuration as for the written news. The average sizes of the three systems are shown in Table 2.

System	Description	Words	Chars
Topiary	Combination of UTD and H. Trimmer	10.5	69.8
H. Trimmer	H. Trimmer alone	9.2	52.2
UTD	UTD topics alone	9.6	68.3

Table 2: Systems and Average Summary Lengths for Broadcast News

9. Evaluation Results

In this section we present ROUGE and BLEU evaluations for written and broadcast news.

9.1 Evaluation on Written News: ROUGE

The ROUGE scores for the three systems and the baseline are shown in Table 3. Under ROUGE-1 Topiary scored significantly higher than Hedge Trimmer, and both scored significantly higher than the UTD topic lists with 95% confidence. Since fluency is not measured at all by unigrams, we must conclude that the Hedge Trimmer headlines, by selecting the lead sentence, included more or better topic words than UTD.

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W-1.2
Topiary	0.24914	0.06449	0.02122	0.00712	0.19951	0.11891
Baseline	0.22136	0.06370	0.02118	0.00707	0.11738	0.16955
H. Trimmer	0.20415	0.06571	0.02527	0.00950	0.18506	0.11127
UTD	0.15913	0.01585	0.00087	0.00000	0.13041	0.07797

Table 3: ROUGE Scores for Written News sorted by ROUGE-1

The highest scoring UTD topics tend to be very meaningful while the fifth and lower scoring topics tend to be very noisy. Thus the higher scores of Topiary can be attributed to including only the best of the UTD topics while preserving the lead sentence topics. The same groupings occur with ROUGE-L and ROUGE-W, indicating that the longest common subsequences are dominated by sequences of length one.

9.2 Evaluation on Broadcast News: ROUGE

The ROUGE scores of the three systems and the baseline are shown in Table 4. None of the systems score higher on ROUGE than the baseline. The difference between Topiary and the baseline is not significant at 95% confidence. However, as with the written news, Topiary scores significantly higher at 95% confidence than either Hedge Trimmer or UTD alone for N=1. As higher order n-grams are used Hedge Trimmer scores higher than Topiary because it contains a higher portion of fluent text.

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W-1.2
Baseline	0.25620	0.09613	0.04243	0.02230	0.22681	0.13852
Topiary	0.24684	0.07193	0.02413	0.01025	0.20463	0.12530
H. Trimmer	0.19716	0.07358	0.02885	0.01320	0.17722	0.10842
UTD	0.19676	0.02869	0.00247	0.00049	0.15981	0.09747

Table 4: ROUGE Scores for Broadcast News sorted by ROUGE-1

9.3 Evaluation on Written News: BLEU

The BLEU scores for the three systems and the baseline are shown in Table 5. For BLEU-1 Topiary scores significantly better than Hedge Trimmer with 95% confidence. Under BLEU-2 Topiary scores higher than Hedge Trimmer, but not significantly. Under BLEU-4 Hedge Trimmer scores slightly but not significantly higher than Topiary, and at BLEU-3 there is no clear pattern. Trimmer and Topiary score significantly higher than UTD for all settings of BLEU with 95% confidence.

System	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Topiary	0.4368	0.2443	0.1443	0.0849
H. Trimmer	0.3712	0.2333	0.1495	0.0939
Baseline	0.3695	0.2214	0.1372	0.0853
UTD	0.2859	0.0954	0.0263	0.0000

Table 5: BLEU Scores on Written News Sorted by BLEU-1

9.4 Evaluation on Broadcast News: BLEU

The BLEU scores for the three systems and the baseline are shown in Table 6. As with the ROUGE scores there is no significant difference at 95% confidence between Topiary and the baseline. Also Topiary significantly outscores Hedge Trimmer and UTD when N=1 with 95% confidence. Once again, Hedge Trimmer scores better when higher order n-grams are used.

System	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Topiary	0.2888	0.1570	0.0880	0.0530
Baseline	0.2767	0.1666	0.1075	0.0727
H. Trimmer	0.2549	0.1551	0.0962	0.0615
UTD	0.2267	0.0916	0.0297	0.0108

Table 6: BLEU Scores on Broadcast News Sorted by BLEU-1

10. Extrinsic Task

For an automatic summarization evaluation tool to be of use to developers it must be shown to correlate well with human performance on a specific extrinsic task. In selecting the extrinsic task it is important that the task be unambiguous enough that subjects can perform it with a high level of agreement. If the task is so difficult that subjects cannot perform with a high level of agreement—even when they are shown the entire document—it will not be possible to detect significant differences among different summarization methods because the amount of variation due to noise will overshadow the variation due to summarization method.

In an earlier experiment we attempted to use document selection in the context of information retrieval as an extrinsic task. Subjects were asked to decide if a document was highly relevant, somewhat relevant or not relevant to a given query. However we found that subjects who had been shown the entire document were only able to agree with each other 75% of the time and agreed with the allegedly correct answers only 70% of the time. We were unable to draw any conclusions about the relative performance of the summarization systems, and thus were not able to make any correlations between human performance and scores on automatic summarization evaluation tools. For more details see (Zajic et al., 2003).

We propose a more constrained type of document relevance judgment as an appropriate extrinsic task for evaluating human performance using automatic summarizations. The task, *event tracking*, has been reported in NIST TDT evaluations to provide the basis for more reliable results. Subjects are asked to decide if a document contains information related to a particular event in a specific domain. The subject is told about a specific event, such as the bombing of the Murrah Federal Building in Oklahoma City. A detailed description is given about what information is considered relevent to an event in the given domain. For instance, in the criminal case domain, information about the crime, the investigation, the arrest, the trial and the sentence are relevant.

We performed a small event tracking experiment to compare human performance using full news story text against performance using human-generated headlines of the same stories. Seven events and twenty documents per event were chosen from the 1999 Topic Detection and Tracking (TDT3) corpus. Four subjects were asked to judge the full news story texts or story headlines as *relevant* or *not relevant* to each specified event. The documents in the TDT3 corpus were already annotated as relevant or not relevant to each event by NIST annotators. The NIST annotations were taken to be the correct answers by which to judge the overall performance of the subjects. The subjects were shown a practice event, three events with full story text and three events with story headlines.

We calculated average agreement between subjects as the number of documents on which two subjects made the same judgment divided by the number of documents on which the two subjects had both made judgments. The average agreement between subjects was 86% for full story texts and 80% for headlines. The average agreement with the NIST annotations was slightly higher when using the full story text than the headline, with text producing 86% overall agreement with NIST and headlines producing 84% agreement with NIST. Use of headlines resulted in a significant increase in speed. Subjects spent an average of 30 seconds per document when shown the entire text, but only 7.7 seconds per document when shown the headline. Table 7 shows the precision, recall and F_α with $\alpha = 0.5$.

System	**Precision**	**Recall**	**$F_{0.5}$**
Full Text	0.831	0.900	0.864
Headline	0.842	0.842	0.842

Table 7: Results of Event Tracking Experiment

The small difference in NIST agreement between full texts and headlines seems to suggest that the best human-written headlines can supply sufficient information for performing event tracking. However it is possible that subjects found the task of reading entire texts dull, and allowed their performance to diminish as they grew tired.

Full texts yielded a higher recall than headlines, which is not surprising. However headlines yielded a slightly higher precision than full texts which means that subjects were able to reject non-relevant documents as well with headlines as they could by reading the entire document. We observed that subjects sometimes marked documents as relevant if the full text contained even a brief mention of the event or any detail that could be construed as satisfying the domain description. If avoiding false positives (or increasing precision) is an important goal, these results suggest that use of headlines provides an advantage over reading the entire text.

Further event tracking experiments will include a variety of methods for automatic summarization. This will give us the ability to compare human performance using the summarization methods against one another and against human performance using full text. We do not expect that any summarization method will allow humans to perform event tracking better than reading the entire document, however we hope that we can improve human performance time while introducing only a small, acceptable loss in performance. We also plan to calibrate automatic summarization evaluation tools, such as BLEU and ROUGE, to actual human performance on event tracking for each method.

11. Conclusions and Future Work

We have shown the effectiveness of combining sentence compression and topic lists to construct informative summaries. We have compared three approaches to automatic headline generation (Topiary, Hedge Trimmer and Unsupervised Topic Discovery) on both written news and transcribed broadcast news, using two automatic summarization evaluation tools (BLEU and ROUGE).

We have shown that Topiary can be applied to broadcast news. At present Topiary is not performing better than the baseline in this domain. We anticipate that our experience in this domain will mirror our experience with written news, in which the research community did not initially outperform the same baseline.

In order to improve performance it will be minimally necessary to improve lead sentence selection and generalize the Hedge Trimmer algorithm which, at present, is tuned to the linguistic characteristics of written news. We plan to move the Hedge Trimmer algorithm away from applying its trimming rules according to a fixed heuristic, to applying the rules according a scoring function that will be based on annotated data from a similar corpus.

12. Acknowledgements

The University of Maryland authors are supported, in part, by BBNT Contract 0201247157, DARPA/ITO Contract N66001-97-C-8540, and NSF CISE Resarch Infrastructure Award EIA0130422

References

Bikel, D., Schwartz, R., and Weischedel, R. (1999) An algorithm that learns what's in a name. *Machine Learning*, 34(1/3), February

Chomsky, Noam A. (1981) *Lectures on Government and Binding*, Foris Publications, Dordrecht, Holland.

Chin-Yew Lin and Eduard Hovy. (2003) Automatic Evaluation of Summaries Using N-gram Co-Occurrences Statistics}. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Alberta.

J. Ma, T. Colthurst, J. Makhoul, S. Matsoukas, H. Gish, R. Iyer, C.L. Kao, N. Dura, O. Kimball, R. Schwartz, R. Prasad, D. Liu, L. Nguyen, M. Noamany, B. Xiang, D. Xu, J.L. Gauvain, L. Lamel, H. Schwenk, and L.Z. Chen. (2004) Speech recognition in multiple languages and domains: The 2003 bbn/limsi ears system. Montreal, Canada.

Mårdh, I. (1980). Headlinese: On the Grammar of English Front Page Headlines, Malmo.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation," In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, PA, pp. 331-318

Richard Schwartz, Sreenivasa Sista, and Timothy R. Leek. (2001) Unsupervised topic discovery. In *Proceedings of Workshop on Language Modeling and Information Retrieval,* pages 72–77, Pittsburgh, PA.

A. Stolcke (2002), SRILM: An Extensible Language Modeling Toolkit. *Proc. Intl. Conf. on Spoken Language Processing*, vol. 2, pp. 901-904, Denver.

Ian H. Witten and Eibe Frank. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufman, San Francisco.

David Zajic, Bonnie Dorr, Richard Schwartz, and Stacy President. (2004) Headline evaluation experiment results, umiacs-tr-2004-18. Technical report, University of Maryland Institute for Advanced Computing Studies, College Park, Maryland.